Number Representation

Data, Algorithms (Processes)

Data Simple/Naive

Algorithms are complex

Simple, reliable

Data efficient, better

whole numbers

Dec → Binary
Binary → Dec

Positive numbers and negative numbers

Positive

Negative

Container - Register (+0.11101)_2

8 bit

Algorithm/process for adding two numbers represented (Sign-mag)
1. Check the sign

Check the sign

- Yes: diff?
  - Yes: operand 1? operand 2?
    - Yes: sign of result is sign of operand 1
    - No: sign of result is sign of operand 2
  - No: operand 1? operand 2?
    - Yes: arithmetic
    - No: sign result

- No: add the numbers
  - Sign of result is same as sign of operands
  - Sign result

- Alg + Alg - 15 steps
- Can we simplify this? Optimize this? Improve performance?

Sign - mag

2's complement

radix - complement
2's complement (when binary)
2's complement:
positive numbers  sign-mag
negative numbers  2's complement

1011101 + 93
01011101

-46

Binary:

8-bit

1's complement: 11010001 + 1
2's complement: 11010010

int x = +93;
int y = -46;
Sign is automatically determined
Result is automatically in 2's complement if it is negative

 overflow